The Pennsylvania State University (PENN, or PSU)
Monte Carlo Code: Basic Description

The Pennsylvania State University (PENN, or PSU, for short) Monte Carlo code is best described by the description provided by Marchuk et al. (1978) in their section 2.2. Since this book is no longer, or never was, in wide circulation in English speaking countries, we reproduce here the first part of their section 2.2:

"2.2 Outline of the Simulation of the Transfer Process

The Monte Carlo method for solving problems of transfer theory consists of computational simulation of photon trajectories according to the following scheme:

1) The original position is simulated by a source distribution $\Psi(r, \omega)$.
2) The free-path length ℓ is simulated.
3) The escape from the medium is examined.
4) The coordinates of the new collision are calculated:

$$x = x' + a\ell, \quad y = y' + b\ell, \quad z = z' + c\ell$$

5) The type of collision (absorption or scattering) is simulated.
6) The cosine of the scattering angle μ is simulated.
7) The coordinates of a new direction are calculated:

$$a = a'\mu - (b' \sin\phi + a'c' \cos\phi)[(1 - \mu^2)/(1 - c'^2)]^{1/2}$$

$$b = b'\mu + (a' \sin\phi - b'c' \cos\phi)[(1 - \mu^2)/(1 - c'^2)]^{1/2}$$

$$c = c'\mu + (1 - c'^2) \cos\phi[(1 - \mu^2)/(1 - c'^2)]^{1/2},$$

in which ϕ is the azimuth of the scattering see Fig. 2.1. ϕ is isotropic, so we have $\phi = 2\pi\alpha$. Here and subsequently, α denotes a random number uniformly distributed between 0 and 1. However, more effective is a rejection method in which $\cos\phi$ and $\sin\phi$ are simulated as the coordinates of the isotropic unit vector,
i) $W_1 = 1 - 2\alpha_1$, $W_2 = 1 - 2\alpha_2$
ii) $d = W_1^2 + W_2^2$; if d > 1, go to i);
iii) otherwise $\cos\phi = W_1 d^{-1/2}$, $\sin\phi = W_2 d^{-1/2}$
8) Go to step 2)

If escape occurs in 3) or absorption in 5), then 1 is stored and a new trajectory is simulated. The most complicated step is the simulation of the free-path length if σ is not constant for the medium. By integration of (2.4) we obtain the distribution function of ℓ,

$$F_\ell(t) = 1 - \exp[-\tau(t)], \quad t > 0.$$

Thus ℓ could be obtained from

$$\tau(\ell) = \int_0^\ell \sigma[r(t)]dt = -\ln\alpha,$$

which is easy to solve if $\sigma[r(t)]$ is a step function, by use of an algorithm similar to that used for $F(\xi) = \alpha$ where the distribution function F is a step function. Therefore the medium is usually divided into sufficiently small regions each having constant σ."

For the I3RC exercises the source is the sun so we input downward moving photons, or energy bundles, randomly across the top of each simulated atmosphere. The simulated atmosphere was discretized into grid cells for which σ was indeed constant, according to the layout of the particular exercise. For each grid cell in which there was the possibility of a scattering event, the corresponding scattering phase function was divided into 20 000 equal probability bins across μ from 1 to -1. For each absorption event simulated photons were not terminated. Rather, a weight p_{wt} of 1 was initially assigned to the photon and each absorption event leads to a decrement of the weight given by $p_{wt}(1 - \varpi)$, where ϖ is the single-scattering albedo associated with the constituents of the grid cell in which the collision takes place.

Irradiances across any surface in the simulation are obtained by taking into account the weight p_{wt} of the photon when it crosses the surface together with the energy associated with the photon, or energy bundle. Contributions to the radiances that were obtained for each exercise are computed for each collision by combining the weight p_{wt} with the probability of escape from the medium in the specified direction. These are standard techniques in common practice in most Monte Carlo simulations.

Two features of the PENN (or PSU) Monte Carlo code are that it is currently parallelized using the P4 parallel-processing library and the input-output to the code is general allowing any type of simulation to be performed. Since the I3RC exercises were the first for which we studied performance issues, many changes were made to the Monte Carlo kernel in order to enhance code speed for both radiance and irradiance calculations. These modifications were sufficiently time consuming that they prevented us from running large numbers of photons for each I3RC exercise. Consequently, for each of the STEPCLOUD, MMCRCLOUD, and LANDSATCLOUD simulations we ran only 3 200 000, 3 200 000, and 8 192 000 photons, respectively. Therefore, we expect the domain averaged irradiances and radiances to be fairly accurate, but the irradiances and radiances (especially the radiances) will be quite noisy for each grid cell in all of the simulations (and especially for the LANDSATCLOUD simulations).

In the I3RC exercises we computed the upwelling irradiances at cloud top; however, we did not compute the downwelling irradiances at cloud base. Rather, we computed the downwelling irradiances at the surface. This difference from the I3RC protocol causes our reported irradiances below the simulated cloud to be smoother than those from the other models. However, since there is not an intervening atmosphere between cloud base and the surface in these I3RC exercises, our domain averaged irradiances and radiances are not affected by this difference from the I3RC protocol.